XG Working Group

The XG Architectural Framework

Request For Comments

Version 1.0

Prepared by:

BBN Technologies
Cambridge, Massachusetts
U.S.A

1 Purpose

This document describes the architectural framework for the development of XG. It also summarizes the main features of XG protocols, interfaces and policies, each of which are detailed in separate documents. This document may be thought of as the "overview" document for the XG protocols. Specifically, the following are addressed:

- Requirements of XG protocols
- Where and how XG functionality fits within a typical networking system
- Layering issues
- Functional decomposition into modules, and interfaces
- Summary of protocols, APIs and policies

This document is a Request For Comments (RFC). Accordingly, an important purpose of this document is to obtain feedback from the community at large, and to refine the ideas described here based on that feedback. The development of the XG architectural framework is an evolutionary process, and this document reflects a snapshot in that evolution.

A number of other RFCs related to XG exist, or are being planned. The complete XG family will consist of the following:

1. *XG Vision RFC*. This lays out the motivation for XG and its scope, presents the key concepts underlying XG, and describes an approach for defining XG.

2. *XG Architectural Framework (AF) RFC*. This document.

3. *XG Protocol RFCs*. Each XG protocol RFC specifies an abstract behavior, and when appropriate, details of individual protocols.

4. *XG Policy RFC*. The XG policy RFC describes the syntax of a policy specification meta-language and a set of example policy specifications.

5. *XG Interface RFCs*. We currently envisage two interface RFCs – XG Transceiver API and XG Opportunity API. These detail the primitives corresponding to these interfaces.

We recommend that this document be read after the XG Vision RFC and before any of the other RFCs.

An important purpose of this document is to articulate, at the highest level, a viable solution that addresses the needs outlined in the XG Vision RFC. In turn, this document is the blueprint to be followed by the protocol, policy and interface RFCs, each of which details a particular aspect of the architecture.

There are, of course, several ways of approaching and defining an XG architecture that realizes the XG vision. Furthermore, the near-term needs are somewhat different from the longer-term needs. In the near-term, the emphasis should be on simplicity and in the longer-term on completeness and lasting value. Therefore, in this document, we have outlined two architectures – near-term and long-term. These will be refined in parallel and merged at a future date – that is, the near-term architecture will evolve and be subsumed within the longer-term architecture to result in a single architecture going forward.

2 Requirements

The design of any system should be guided by a clear set of requirements. This section identifies requirements for the XG architecture and protocols. Every aspect of the architecture and protocols should be traceable to one of these requirements, and eventually, every requirement should be supported by some aspect of the protocols.

Requirements include (we don't include obvious requirements, such as "must provide dynamic spectrum access with interference mitigation", which follow directly from the XG vision).

1. It must be possible to add XG to a legacy system. Such an addition should not require extensive modifications to the legacy MAC mechanisms.
2. Legacy systems without XG extensions should interoperate with XG-enhanced systems.
3. There must be a provision to incorporate spectrum policies, priorities, and exclusions into the functioning of the protocols and/or abstract behaviors.
4. The XG protocols must be largely agnostic to the MAC layer technologies. They must not depend upon how a MAC layer functionality is implemented. The general behavior of an XG system should be largely independent of the nature of the MAC layer, though a particular XG implementation may be aware of and exploit particular MAC layer technologies.
5. The XG protocols must be largely agnostic to the physical layer technologies. That is, it must not depend upon how a physical layer functionality is implemented. The general behavior of an existing XG system should be largely independent of the nature of the physical layer. A particular XG implementation may be aware of and exploit particular physical layer technologies.
6. A core set of behaviors must be identified in such a manner that a viable architecture where only the core set needs to be considered for regulatory approval is possible.
7. The framework and protocols should be flexible enough to support XG-like capabilities long after the initial DARPA XG implementation(s).

A major goal of the RFCs is to present an *abstract* view of XG. In particular, the problem statement is not with respect to any one existing protocol, nor will the solution be simply an embellishment of an existing protocol, such as 802.11. While it is likely that such embellishments will prove useful for the initial implementations of XG, the RFCs themselves will be at "one level higher" and solve the *generic* XG problem.

Another goal is to keep the core behaviors distinct from the innovations that may implement the mechanisms in different ways. This would be analogous to secure kernels – that is, inside the boundary, we can be sure of what is happening and can trust it whereas outside this boundary there is room for innovation. The challenge is to make it so that only the core set of behaviors "inside the boundary" is relevant for regulatory approval.

3 Preliminaries

The description of the XG architecture will employ several concepts, such as layers, modules, interfaces, behaviors, and interference and XG domains. In this section, we define these concepts precisely so that all readers may interpret the rest of the document uniformly.

3.1 Layering

A *layer* is a level of abstraction that captures some important aspect of the system, provides an interface that can be manipulated by other components of the system, and hides the details of how the encapsulated functionality is implemented. A *protocol* is a set of rules governing the format and meaning of messages that are exchanged by the peer entities at the same layer. A protocol provides a communication service that higher-layer objects use to exchange messages. The basic idea of *layered protocols* is for a lower layer to provide services to the layer above it.

A layer can be divided into smaller logical substructures called *sublayers*. Sublayers abstract functionality within a layer in just the same way that layers abstract functionality within a system. Clearly, this definition can be generalized recursively.

When two adjacent layers (or sublayers) differ in some aspect of functional perspective, and we need to make the differences transparent to the higher layer (or sublayer), the concept of a *convergence layer* or “adaptation” layer is useful. A classic example is the layering of IP over ATM, where the variable-length IP packets need to be segmented into fixed length ATM cells. A convergence layer is introduced that does segmentation and reassembly. The convergence layer is similar to a layer, but is used when a simple, specific functionality is targeted in relation to “mapping” between two adjacent layers. In other words, there is no new significant functionality introduced, but a kind of “translation” happens. In our case, the diversity of technologies at the physical and MAC layers and the need to map between them motivates the use of convergence layers.

3.2 Modules and Interfaces

Layers, sublayers, and convergence layers are connected by *service interfaces,* or *APIs* (we use the latter term). An API provides a set of *primitives* using which the service provided by a (convergence/sub)layer can be suitably abstracted. Use of APIs provides a means for modular development of system components, perhaps by different performers. APIs between layers/sublayers are termed *vertical APIs*. In contrast, *horizontal APIs* are interfaces between modules at the same layer.

3.3 Domains and Regions

The application of XG principles and techniques is concerned with maintaining predictable levels of interference among potentially competing radio systems. We distinguish between the radio systems themselves and the locations in space that they occupy with the following definitions.

The terms *set* and *domain* refer to collections of individual radio systems (sometimes called *nodes*). The term *region* refers to a geographic area in which one or more nodes may be located.

There are different types of sets, domains and regions.

The i*nterference set* of a node consists of all nodes with whom that node may interfere. An *interference domain* is the transitive closure of the interference sets of one or more nodes.

An *interference region* is the contiguous area occupied by an interference domain, extended to include the area that would be occupied by “would-be” interferers with members of that domain. That is, the area subject to the interference of/by members of the interference domain.

An *XG domain*, on the other hand, refers to a collection of nodes that are able to exchange opportunity information for the purpose of making choices about the use of spectrum or other transmission-related resources. Members of an XG domain are able to cooperate in the utilization of spectrum-related resources.

3.4 Channels and Opportunities

We assume that the operational spectrum for XG can be partitioned into non-overlapping *channels*, which is the fundamental “unit” of spectrum for dynamic management. For instance, a 100 MHz band could be partitioned, for XG purposes, into 10000 channels of 10 KHz each. It is not necessary to have channels of equal width. Properties (such as presence of a primary signal) are determined on a per channel basis. A channel is the smallest unit for which such properties can be described.

An *opportunity* exists if an XG node can transmit using some combination of operating parameters such that existing primary nodes (if any) do not perceive interference, for a given threshold of such interference. We note that the definition of the opportunity is node and threshold dependent (amongst other things), and so the same “sensing” information may or may not represent an opportunity.

3.5 XG Nodes

The XG nodes and the networking context were discussed in detail in the Vision RFC. Here we reiterate the terminology for the different kinds of nodes that play a role in the architecture.

- *Non-XG.* These are "traditional" non-XG-capable nodes, or are running a different (incompatible) set of protocols. Our XG protocols must be interference preserving with respect to such nodes under the assumption that they are operating legally.

- *XG-aware.* These are nodes that can exchange information about what frequencies they are using and may make use of information about the frequencies that our node/network is using. However, they do not cooperate in terms of frequency assignment. Our XG protocols must find out and avoid frequencies used by such nodes, and should inform them about the frequencies we use.

- *XG-cooperative.* These are nodes that can use the XG protocols to coordinate the use of spectrum. They run an interoperable implementation of the XG protocols. These are nodes with which distributed dynamic spectrum sharing typically works. We often call these simply *XG nodes.*

3.6 Abstract Behaviors

Finally, we discuss the concept of *abstract behaviors*. An abstract behavior is an abstraction of a protocol that hides details of one or more aspects of its functionality. This hiding could be done at several levels, and so one could have several levels of abstract behaviors. For instance, consider the IEEE 802.11 MAC Distributed Coordinated Function. A protocol for this involves specifying the frames (RTS/CTS/DATA/ACK), their formats (waveforms), timers, finite state machines, and so on. A first level abstract behavior might be to simply say "… must use RTS/CTS/DATA/ACK handshake for collision avoidance…". This behavior might be implemented by a variety of protocols that might differ in packet format or how the NAV is handled. An even higher level abstraction might be to say "… must avoid collisions…" allowing different kinds of algorithms, including TDMA. For XG, we will choose appropriate levels on a per protocol basis, based on standardization and regulatory issues.

4 XG Framework

The XG system is a complex one. Architecting complex systems has long been recognized as a problem that is best addressed using a formal framework. One of the approaches that has become popular is the Zachman Framework for Enterprise Architecture. In this section, we adapt the ideas behind the Zachman Framework to present an XG framework. Such a framework is helpful in providing a panoramic view of XG that puts in perspective the material presented in this and other RFCs.

4.1 The Zachman Framework

The key idea behind the Zachman Framework is that a complex system may be viewed at different levels, by different "players". The classic example, and one that was introduced in the original paper by Zachman [Zach], refers to the construction of a house. In this, the owner often gives broad requirements for the house; the architect prepares architect's drawings that depict the house from the owner's perspective. These are then made into formal plans that are a designer's representation of the final product which would be used by the contractor, who in turn makes detailed engineering plans to be used by the builder, and so on. Each of these levels is important and used by a particular person in the development process.

The crux of the Zachman framework is a matrix where each row represents a different *perspective* of the system. These views include contextual (scope), conceptual (business model), logical (system model), physical (technology model), and detailed representations. Each column represents a different *description* of the system. Descriptions include what (data), how (function), where (network), who (people), when (time), and why (motivation). Thus, each cell in the matrix captures a unique aspect of the system, and is explictly differentiable from all other cells in the matrix.

The Zachman Framework does not indicate a methodology for filling out the cells, nor does it offer any guidelines on what should be done with the matrix. Rather, it appears to be a tool for enhancing clarity of thought, and as a structure that ensures that all aspects of the system are covered.

4.2 XG Framework

Like many other complex system, XG also needs to be understood at many "levels", from conceptual to detailed implementation. Each "player" in the development of the XG system has a different view of XG, according to his or her requirements. For instance, the FCC has a view that is largely focused on regulatory issues and on ensuring interference preservation, which is quite different from the system integrators view which needs to consider performance and other aspects. All of these different views are important. It is helpful at this stage to consider the broadest picture of XG and look at these different views, so that each RFC, including this one can be put in perspective.

The Zachman Framework was designed for an enterprise, and XG is not an enterprise. Thus, it is not directly applicable for our purposes. However, the key idea behind the framework – namely, that there are different perspectives and descriptions – may be taken and adapted to our context. Such an adaptation is presented in Table 1 below. We have changed the semantics of the rows slightly and considered only the three most relevant items in the descriptions.

The XG program benefits from such an architectural framework because the operational views provide a means for communicating with high-level decision makers, e.g., DoD, commercial and FCC executives. The conceptual and design views allow the design of core behaviors that can be used by the high-level decision makers for regulatory purposes. The system views allow engineers to understand the XG system functions and interfaces, and the technical views show how XG fits into the existing standards and how it adapts to changing standards in the future.

	WHAT	WHY	HOW
Scope (DoD/FCC)	♦ Dynamic spectrum management	♦ Increased capacity ♦ Better use of spectrum ♦ Zero setup time ♦ Regulatory simplicity	♦ Build a system that uses unused frequencies in an interference preserving way
Concept (XG PM)	♦ Abstract (core) behaviors ♦ Protocols ♦ Policy language	♦ Long-term impact ♦ Regulatory approval ♦ Flexibility	♦ RFC process ♦ Performer participation ♦ Industry feedback ♦ FCC/DSO involvement
Design (Contractor: BBN and working group)	♦ Architectural framework (near and far). ♦ Behavior specs – sensing, identification, allocation, use.	♦ Long-term impact ♦ Regulatory approval ♦ Flexibility	♦ AF RFC ♦ Behavior RFCs ♦ Policy RFC ♦ API RFC ♦ WG interaction
Technology (Contractors: SS, Raytheon, LM)	♦ Sensing algorithms ♦ Transmit power estimation behaviors ♦ Allocation algorithms ♦ Morphed waveforms ♦ Tieing up with existing protocols (e.g 802.11) ♦ Component technology	♦ Interference preservation ♦ Noise temperature control ♦ Backward compatibility with legacy ♦ Future upgrades in a competitive manner	PROPREITARY

Table 1

5 XG Layering Issues

In our vision, XG is implemented as a layer 2 process as shown in Figure 1. A legacy stack (left) may be upgraded to use XG without modification to the legacy MAC protocol. The legacy Transceiver API may be

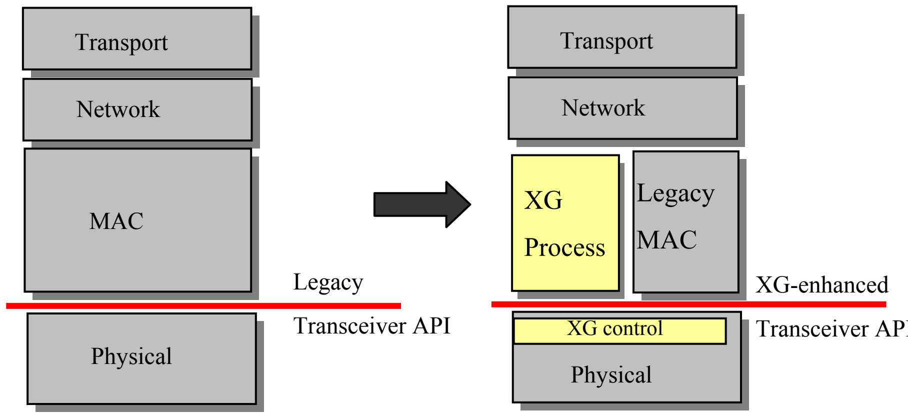


Figure 1 - XG Process in Stack Context

enhanced to include certain XG-specific primitives to provide an XG-enhanced Transceiver API as shown in the figure (right). Note however that this does not require a change in the legacy MAC protocol as it continues to use the subset of the API that it originally used. Thus, the legacy MAC need not be aware of XG. There is no change to the network layer and above – the scope of XG is entirely restricted to physical and MAC layers.

The physical layer implements a minimal "XG control", in that it recognizes that some of the MAC requests may imply XG-specific action. The XG process communicates with peer XG processes at other nodes to exchange spectrum information, and other XG control information (Figure 2).

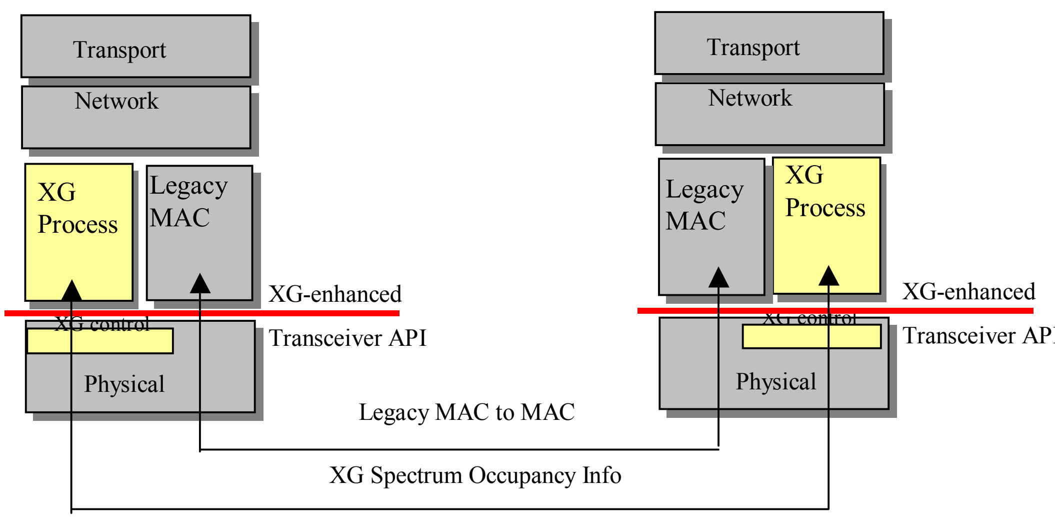


Figure 2: XG peer interaction

We note that the XG process being at layer 2 is with respect to *application data* packets. This does not preclude XG *control messages* from reusing some of the higher layer functionality – for instance, using TCP to communicate to a peer XG process in the same domain. We do not consider this a layer violation since the layering of functionality applies to data packets[1]. Our stack model does not imply that routing, transport, encryption has to be re-implemented at layer 2.

[1] This is similar to the use of TCP for control message delivery by BGP (Border Gateway Protocol) which is considered to be a layer 3 protocol.

The XG processes coordinate with each other to implement a dynamic spectrum sharing procedure amongst themselves in a manner that is designed to control interference to existing primary users. The XG process then creates the "state" in the physical layer for appropriate handling of packets consistent with decisions made in the XG process. This is achieved by certain control primitives implemented as part of the XG enhancement of the Transceiver API. For example, the XG process might have determined that frequency channels *f1* through *f2* may be used by this node at this time. MAC packets then are transmitted on these channels. The state may also contain instructions on any modifications that may need to be made to the outgoing and/or incoming packets on behalf of XG. In other words, the XG control at the physical layer executes data packet processing on behalf of the layer 2 XG process[2].

The XG process utilizes the physical layer to communicate and exchange spectrum utilization perceptions, and then to coordinate frequency assignments for the radios in the physical network. This exchange is essential because we need to both ensure that the selected frequency is usable at the receiver, and is not likely to jam signals from the environment of the transmitter.

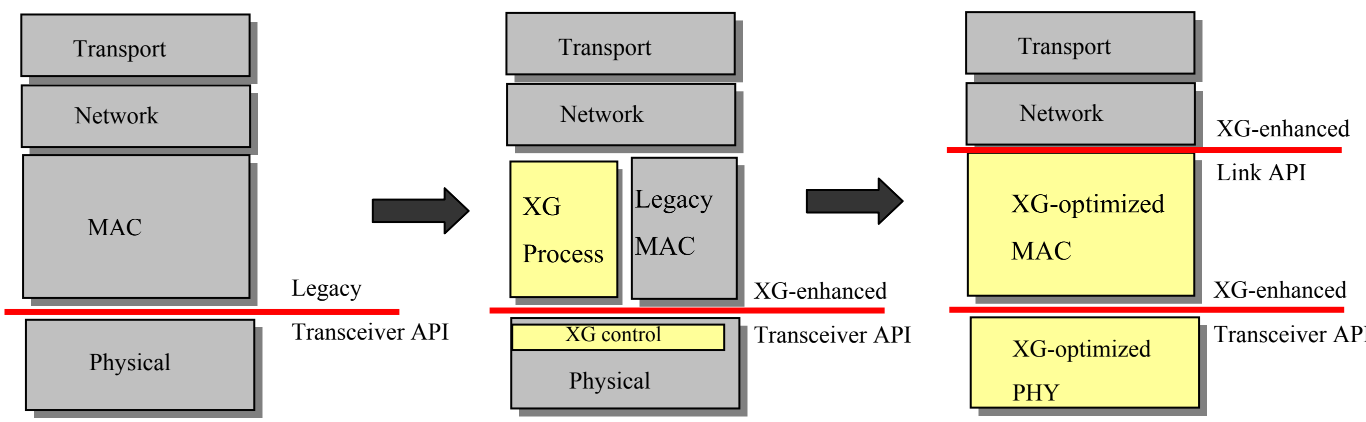


Figure 3 - Evolution from Legacy, through co-existing XG/Legacy, to Optimized XG

This approach has the advantage that it allows the use of XG with existing MAC designs. In other words, it should be capable of supporting legacy MAC code before the transition to XG-Optimized MAC and Physical layers is made. The transition from current day practice to "full XG" is shown in Figure 3.

In some cases, it is likely that the XG protocols cannot use the native physical layer. This is true when this layer has unique characteristics, or when there is no common mode of operation, such as in linking heterogeneous networks. It is also true in cases where the XG protocols require an appropriately XG-enhanced physical layer. In the most extreme case, one of the XG-enabled systems may not be a communications system at all, but may be a sensor that communicates and/or coordinates spectrum usage with communications or other sensor systems. In such cases, we perceive that we will need to define a physical layer standard for an XG interoperability path, which can be selected as part of the XG standard, or negotiated among the radios. Since Software Defined Radios are the likely implementation platforms for XG, the introduction of an additional physical layer is not as significant as it would be with discrete implementations, but is still a complexity we would like to avoid. Some means of determining a common mode of operation could be a more suitable solution. This is a technology that DARPA is investigating in other programs, and may remain outside of the current XG work.

The above representation is only the simplest form of XG. Clearly there are very significant benefits to the system's ability to be aware of, and to utilize network topology information that is only accessible in the upper layers, such as the membership data that likely resides in the network layer. We will be investigating these, and similar, opportunities for enhanced performance later in the program. We intend, if possible, to develop this functionality in the context of the same set of abstract behaviors that are used in the core architecture. We envision that with the above mentioned and other features, an "ultimate" XG architecture would have XG

[2] This avoids having to actually pass the data packets "up the stack" to the XG process – doing so is a "layer violation".

optimized MAC and physical layers with the network layer being made aware of XG features by means of an API (which could be used to supply network topology information, for instance). This is illustrated in the rightmost diagram in Figure 3.

6 XG Modules

In this section, we present a first-level modular decomposition of XG functionality. We begin by recalling the broad architectural vision presented in the companion document "XG Vision RFC", and then work our way through a decomposition.

We shall use the rightmost diagram in Figure 3 as a starting point for the decomposition as it is the most general, and also allows us to focus on the XG functionality. Note that this choice does not in any way detract from the vision of having the legacy MAC and XG MAC coexist without changes to the legacy system. To see this, simply imagine that a legacy MAC box is placed alongside the XG-optimized MAC box in the rightmost diagram, and note that the XG-enhanced Transceiver API contains all of the primitives that the legacy system used (since it is an "enhancement"). Now, having affirmed this, we ignore the legacy MAC in order to concentrate on XG. Thus, without loss of generality, we use the rightmost diagram as the conceptual basis for the development of the architecture.

We now consider a first-level decomposition of the XG MAC and physical layer functionality. XG is mostly a MAC level system, however, some of the key pieces it requires are arguably at the physical layer[3]. One example is *sensing* – the collection, and possible averaging of received signal strengths, perhaps over a wide bandwidth. This requires some consideration of *cross-layer* issues. A goal is to cleanly manage such issues using the XG Transceiver API.

The modular decomposition is given in Figure 4 below. There are three high-level modules: *Opportunity Awareness, Opportunity Allocation,* and *Opportunity Use*. We define their functions briefly below, and elaborate them later.

- *Opportunity Awareness*. This determines the set of available opportunities and associated constraints on their use. This set is dynamic, that is, changes as a function of time. The opportunity availability is determined for a subset of XG nodes, typically in the neighborhood (within a certain radius) of the given node. The opportunity awareness function is a distributed procedure that may include any or all of the following – sensing of spectrum opportunities, identification of usable opportunities and associated constraints, and the dissemination of this information to an appropriate neighborhood.

- *Opportunity Allocation*. This is a distributed procedure that allocates the available opportunities (as determined by the opportunity awareness module) for transmission amongst the XG nodes. The allocation is dynamic, that is, changes with time. The opportunity allocation may be done based on any medium access control approach – CSMA/CA, FDMA, TDMA, CDMA, or a combination thereof. Clearly, the mechanism depends upon which approach is used, but the functionality itself is agnostic to the actual mechanism. However, the mechanisms that can be used depends upon the awareness information available (for instance, if no code opportunity information is available, one cannot exploit code opportunities and allocate them).

- *Opportunity Use*. This refers to the physical layer mechanism that achieves transmission of a set of packets over the set of indicated opportunities. That is, its job is to ensure that a packet is transmitted as quickly as possible subject to constraints supplied to it (e.g., transmit on frequencies *f1-f7*, at power level not to exceed *p*, and a spreading of at least *k* chips/bit). Clearly, a large number of possible opportunity use mechanisms exist, from sequential channel access to morphed waveforms. Again, the module does not dictate how it should be done, merely what is to be done.

[3] One may ask: why do we need to bother about which layer something is? Why not simply treat the entire XG system as one big "box" and think about a decomposition. This is possible – in the same way that it is possible to think of a router without using layers. The fact is that layering allows conceptual clarity and a certain relationship to existing functions which is helpful.

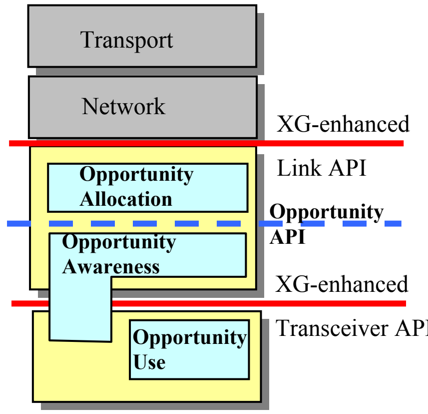


Figure 4: XG Modules in a layering context

Figure 4 also shows an API – the opportunity API – between the allocation and awareness modules. This API helps to cleanly separate two functions – determining opportunities, and using them. This allows independent progressive refinement of each, and a large number of solutions for each within the same framework. Although it is an XG-internal API, it is a significant one because it separates the behaviors that are likely to have regulatory implications from those that will largely be outside of regulatory purview.

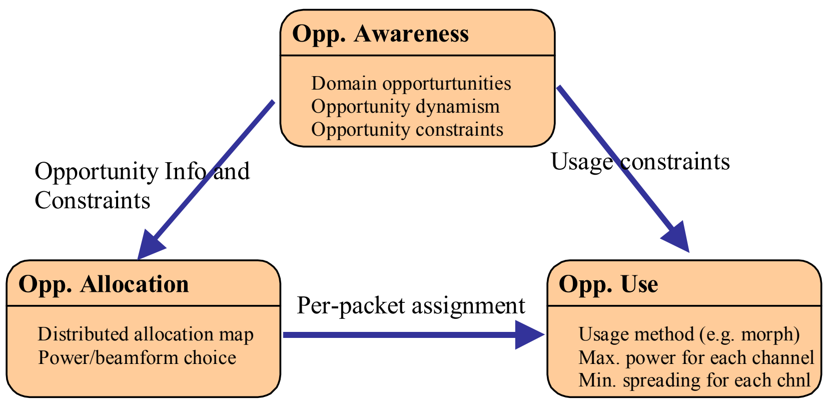


Figure 5: Top-level XG modules

Figure 5 above shows an example of the interaction between modules and an example of information resident in each.

The opportunity awareness module tracks the opportunities in the XG *domain* (refer to section 3.3), and its dynamism, including perhaps information to predict the availability of opportunities. It also tracks the constraints on the opportunities, such as time window, maximum power and other transmitter parameters that need to be used to use that opportunity. The identification of opportunities is controlled by policies.

The opportunity allocation module uses the opportunity information and constraints from the awareness module and creates a dynamic allocation map. The allocation map is essentially a distributed database of frequency, and possibly time slot or code assignments to XG nodes. This module also tracks information such as the power and beamforming to use for that assignment, etc. For a given packet, it then provides the assignment of opportunity to the opportunity use module.

The opportunity use module tracks current preference of usage method (if there are multiple), and maintains a set of lower and upper bounds on transceiver parameters for interference preservation. Such bounds are created using information from the awareness module which can supply usage constraints. Arguably, the allocation module can provide this information too, as it knows (and has to know) the usage constraints. However, the flow of information in the architecture is more streamlined if this information were given directly. Furthermore, this information relates directly to behaviors that are closely associated with interference preservation. By removing this from the allocation module, we allow for the possibility that all of allocation is "non-core", that is outside of regulatory purview. This would be a step in the direction of our having a small set of core behaviors within a boundary and allowing innovation outside of it (see the Vision RFC for more on this).

We emphasize that the information depicted in Figure 5 is only an example. A number of other pieces of information are relevant and will be presented in a more detailed version of the design.

6.1 Opportunity Awareness

We now consider the opportunity awareness module in more detail. This is the key module for XG, and undeniably the most significant from an architectural viewpoint. A natural decomposition of this module is in terms of *sensing, identification* and *dissemination.*

- *Sensing.* This is the process of sampling the channel in order to determine occupancy. We note that there is no fixed definition of when a channel is occupied – it depends upon the receiver (its sensitivity for instance), the sampling window, the average and peak values within the window, thresholds on discriminating noise from signals etc. The criteria for declaring a channel occupied may also change with time. However, the basic notion is to determine if there is a signal, and if so, what the characteristics of the signal are.
- *Identification.* This is the process of determining whether a channel is an *opportunity*. Note that sensing merely tells you the characteristics of the channel. Identification, on the other hand, uses this information to determine whether or not it can be used by XG. If a channel is sensed free, it may or may not be prudent to use it (maybe we are in a deep fade). Similarly, even if a channel is occupied, it may be acceptable to transmit within a power level. Thus, identification contains the algorithms to convert sensed information to be useful to XG.
- *Dissemination.* As mentioned earlier, opportunity awareness needs to include not just the node but also some subset of its k-hop neighbors. This is because, allocation mechanisms often do much better with somewhat global knowledge. Dissemination is the process of distributing the information to other nodes so that opportunity awareness to the extent necessary is achieved. The phrase "to the extent necessary" captures a whole slew of possible mechanisms, each interacting with a possible allocation mechanism. This is deliberate, and is an example of the range of innovations possible and the need to support such innovations in the regulatable kernel.

Figure 6 below shows an example of the interactions between the submodules of opportunity awareness. As in Figure 5 the information held by each and the interaction between the submodules are also depicted.

The sensing submodule tracks the signal level characteristics in the channel, perhaps even identifying it as primary or secondary. It might also keep track of the activity statistics, or the most recent history of activity. If the sensitivity threshold can be adjusted, it may store the current value used.

The sensing submodule provides the identification submodule channel activity information to help determine whether or not it is an opportunity. At some logical, goal-specific level, the information that flows from sensing to identification is the set of *possible* opportunities. The identification submodule determines which of these are *real* opportunities, and how they can be used. Accordingly, the marking of a channel as an opportunity is tracked by the identification submodule, as is the expected lifetime of the opportunity and constraints on its use (such as maximum power).

The "real" opportunities are then passed to the dissemination submodule that is responsible for collecting the local opportunity information at various nodes. Accordingly, it maintains the local opportunity information as well as opportunity information for the relevant sub-domain in the XG node's neighborhood, and related constraints. This is the data that is provided to the opportunity allocation module. Dissemination takes time and network resources. Therefore, the architecture supports multiple maps of the relevant sub-domain, for instance, an accurate (up-to-date) view of coarse granularity and an approximate (out-of-date) map of fine granularity. Different allocation schemes may want to use these different levels of information.

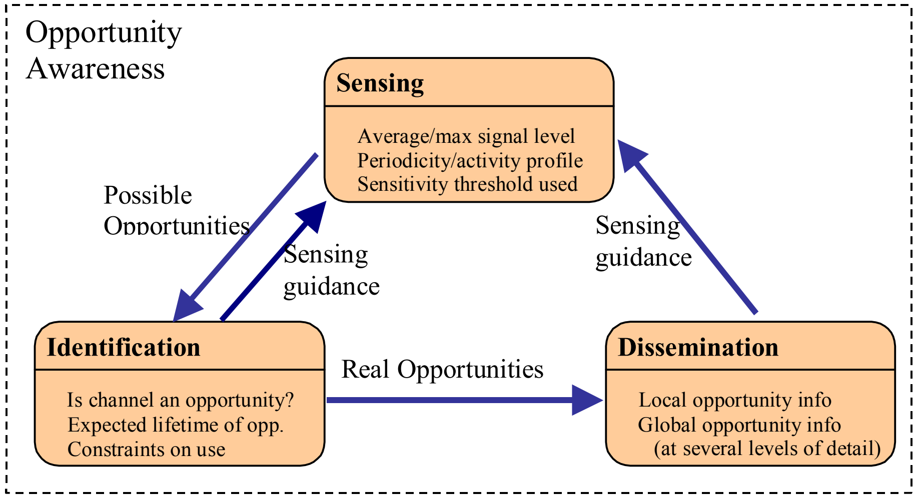


Figure 6: Submodules of Opportunity Awareness

Finally, the identification and dissemination processes may provide information to the sensing module to enable better or more efficient sensing. For instance, if it is determined that a channel is likely to be an opportunity for the next several hours, or if it is determined that a channel should be left alone for a period of time, the sensing module can skip that channel and save time.

We emphasize that the information depicted in Figure 6 is only an example. A number of other pieces of information are relevant and will be presented in a more detailed version of the design.

7 XG Architecture: Near-Term Usage Examples

Recent measurements have shown that a typical geographical region has wide swathes of spectrum where there are no users at all. Thus, even without sophisticated predictor-corrector or dynamic management techniques, one can dramatically improve system capacity and enable rapid entry into an area without apriori frequency assignment.

The "near term usage examples" in this section is aimed at plucking such "low hanging fruit". Another way of looking at the goal of this architecture is: what is the simplest set of techniques/protocols for opportunistic use of spectrum? In particular, the goal here is not the *optimum* use of resources, but the *easiest* way to reasonably utilize gaps in spectral occupancy. XG must be able to function along with existing technologies without requiring any modification of them. XG functionality must be inserted as unobtrusively as possible into current architectures. The near-term usage examples are based on the middle diagram in Figure 3. We will refer to this as the XG *near term* architecture and reiterate it below in Figure 7.

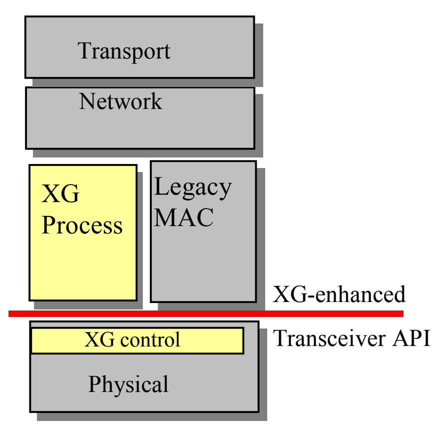


Figure 7

As envisaged in the Vision RFC, the near-term XG architecture allows the co-existence of the XG process with a legacy MAC. Specifically, we consider a CSMA/CA (as in 802.11 a/b/g) MAC and give examples of how the XG process might inject a modicum of dynamic spectrum management using very simple strategies.

In this framework, the XG control detects Legacy MAC (L-MAC) frames that have XG implications. These generate requests through the XG-enhanced Transceiver API to the XG process that triggers certain behaviors in order to satisfy the L-MAC requests. Such behaviors may involve the encapsulation/decapsulation of the L-MAC packets (transparently to L-MAC), or simply the selection of appropriate XG physical layer services for the transmission/reception of the L-MAC packets. The XG process may also exchange opportunity information with peer XG processes to do intelligent, coordinated sensing. It might also use it for disseminating opportunity information for resource allocation purposes.

We illustrate the concept of operations with this architecture using a few examples, identified as *XG-less fallback, Zero control, Frequency selection,* and *Frequency negotiation.* In each case, we identify the peer interactions and packet modifications that might be necessary. For these examples, we assume a 802.11-like CSMA/CA protocol. However, we note that this is only for illustration and the architecture by no means restricts the kind of legacy protocol. Also, we emphasize that these are only examples – the fact that embellishments of 802.11 are suggested does not alter our goal of keeping the architecture, and this RFC at a general level.

XG-less Fallback

When used with other legacy radios, the architecture defaults to a “no operation” with respect to XG functionality as follows. All L-MAC protocol data units (PDUs) are simply sent out without any processing by the XG control. This could happen based on configuration or set as the default case, with XG functionality being invoked only upon discovering the presence of one or more XG-capable nodes.

Zero Control

In this example concept of operations, we assume that the transceiver has wideband tuning capability on receive and frequency agility on transmit. The XG process senses a contiguous set S of channels that are completely unoccupied in that geographical area. When receiving, the node listens on all of the channels in the contiguous set S and when detecting a packet on one of the channels, it tunes to that channel. When transmitting, XG picks one of the channels at random and sends the packet. The state for this behavior is placed in the XG control by the XG process, so that the packet does not have to travel back up to the MAC layer. The RTS/CTS/DATA/ACK packets are sent as-is, except that a selection of channel from among those unoccupied is made via the XG Transceiver API.

This could allow multiple parallel communications on different channels to occur within a geographical area in the simplest possible manner. We note that the legacy MAC, for example 802.11, can be completely oblivious of the XG functionality.

Several behaviors are possible upon collision of, say the RTS, due to two nodes picking the same channel. The XG process could simply not take any action, but rely on the L-MAC retransmissions (which might result in a new unoccupied channel being picked by the random process). Or the XG process could, through the XG control, itself attempt retransmissions, or proactively send multiple RTSs on different random channels. We note that some of these behaviors may have interactions with a legacy MAC. For instance, if XG attempts its own retransmissions, an XG-unaware L-MAC would timeout on the RTS.

Frequency Selection

This assumes the existence of an a priori dedicated control channel that is known to all nodes. The RTS/CTS are sent on the control channel and used to select a channel for the DATA/ACK. The XG process identifies the set of possible channels by sensing. There is no need for the channels to be contiguous. There is also no need for the transceiver to be wideband tunable, but it should be frequency agile. When in idle, the transceiver is tuned to the control channel.

In the context of the architecture, the operation is as follows. The XG control encapsulates the RTS into a new packet, say, X-RTS. The X-RTS contains the suggested channel number *c* for the DATA/ACK communications, chosen randomly from among those available. The peer XG control of the receiver decapsulates the X-RTS and sends it to its L-MAC. It also notes whether the channel *c* is usable or not. If it is, then the corresponding CTS is encapsulated into an X-CTS, conveying that this channel is fine. For the DATA and ACK, no encapsulation is necessary, but the physical layer is directed to select the channel *c* (similar to zero control). Peer XG control modules of nodes that receive the X-RTS or X-CTS simply decapsulate and pass it to their L-MACs that perform the usual NAV operations. Additionally, they also note which channels have been chosen or in the process of being chosen, so that they can avoid those channels for selection to put into the X-RTS or X-CTS.

Once again, we note that the architecture allows for the above concept of operations to happen without the legacy MAC being XG-aware.

Multiple Frequency Negotiation

This is similar in spirit to the frequency selection, except that multiple frequencies are chosen, and there is scope for the receiver to pick a subset of the frequencies it is offered. As in frequency selection, we assume the existence of an a priori dedicated control channel that is known to all nodes, transceiver should be frequency agile, and we assume that sensing is done independently by the XG process. The identified frequencies may be non-contiguous. The frequencies negotiated for communication may be contiguous or non-contiguous – if the latter, then it is assumed that the physical layer has the capability to send a packet over non-contiguous frequencies.

In the context of the architecture, the operation is as follows. The XG layer encapsulates the RTS into X-RTS, listing a number of possible channels that can be used. The X-CTS is returned with a subset of these channels, and the DATA and ACK use one or more (if non-contiguous frequencies can be used) of these channels. As before, nodes other than the transmitter and receiver keep track of the chosen channels using the overheard X-CTS.

Once again, we note that the architecture allows for the above concept of operations to happen without the legacy MAC being XG-aware.

This transparency, however, may lead to sub-optimal performance. Consider for instance the case when a morphed waveform is used to transmit the DATA over multiple frequencies, thereby shortening its transmission time. Ideally, this should allow other nodes start their pending transmissions sooner than if the DATA had been sent on a single channel. However, the fact that the L-MAC is XG-unaware means that the NAV period will not be adjusted, and hence the nodes will continue to be backed off for a time equal to if a single channel had been used.

8 Abstract Behaviors

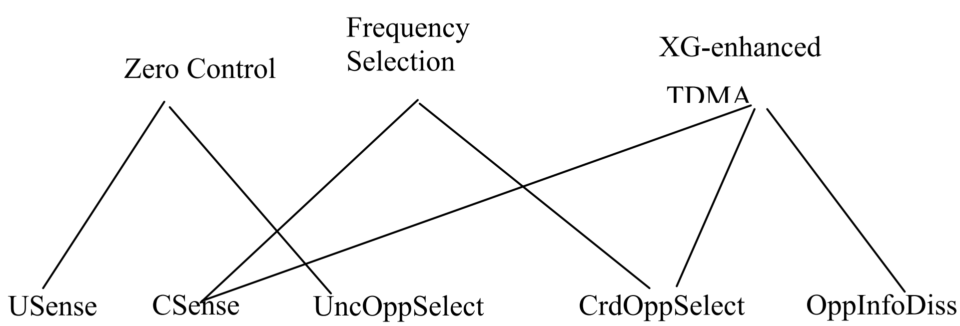
A key goal of the RFC process is to specify a set of abstract behaviors for XG. The first step toward that, of course, is to decide: *what are the abstract behaviors that should be specified?* This section “gets the ball rolling” by identifying a set of possible behaviors. Recall that the specification of the behaviors themselves are part of the protocol RFCs.

Following our modular breakup, we can identify five top-level behaviors: sensing, identification, dissemination, allocation and use of opportunities. Each of these behaviors may be represented as an abstract class with resident data and methods for access. At a second (lower) level of abstraction, we can identify behaviors that correspond to different ways of achieving the desired top-level behavior. These include:

- *Uncoordinated Sensing*. The sensing here is completely local to the node, and the opportunity is identified based solely on spectrum occupancy as seen by this node.
- *Coordinated Sensing*. Control messages are exchanged between nodes in order to implement a “quorum” based decision on opportunities.

- *Uncoordinated Opportunity Selection*. The selection of an opportunity for DATA transmission is done without consultation with the peer(s) involved in the communication. For example, Zero Control employs this.
- *Coordinated Opportunity Selection*. The selection of an opportunity for DATA transmission is done based on one or more handshakes that might involve XG control messages. For example, the X-RTS and X-CTS control messages are used in the Frequency Selection approach.
- *Opportunity Information Dissemination (OID)*. Dynamic opportunity information is exchanged between nodes in a neighborhood so as to help coordinated opportunity selection. This might involve control messages that convey such information over multiple hops so as to make more efficient allocations.

As an example, the mechanisms described in section 7 can be seen as combinations of some of the above behaviors. This is illustrated in figure below.



Each mechanism is appropriate for a different set of hardware and assumptions: e.g., zero control when no control channel, but wideband tuning on listen; multiple frequency negotiation when control channel and ability to spread a packet over multiple non-contiguous frequencies etc.

The total number of possible mechanisms is potentially large, but the set of core behaviors could be much smaller. By itself, a core behavior may not provide all the XG functionality required to dynamically use spectrum. However, if regulatory bodies approve each behavior, then the entire mechanism's behavior is approved. For instance, if one can show that all of the core behaviors are *interference preserving* – that is, the introduction of a signal will not degrade the performance of any then operating system by more than a set threshold – then the mechanisms constructed out of these behaviors will also be interference preserving.

Another advantage of this approach is that given the core behaviors, perhaps the mechanism best suited for the assumptions and hardware can be constructed on the fly.

We note that these are just initial ideas for how to identify the right set of behaviors to specify, and expect them to change as thinking evolves. Abstract behaviors will be specified in more detail in the protocol RFCs.

Acknowledgments

This document was prepared by the Internetwork Research Department, BBN Technologies, with input from DARPA/ATO, AFRL, DSO, MITRE, Raytheon Corp., and Shared Spectrum Corp.

Comments

Comments on this RFC should be emailed to Ram Ramanathan at ramanath@bbn.com, along with the commenter's name and organization.